Visual Place Recognition with Repetitive Structures

Akihiko Torii, Josef Sivic, Masatoshi Okutomi, Tomas Pajdla

Visual Place Recognition with Repetitive Structures

Akihiko Torii, *Member, IEEE,* Josef Sivic, *Member, IEEE,* Masatoshi Okutomi, *Member, IEEE* and Tomas Pajdla, *Member, IEEE,*

Abstract

Repeated structures such as building facades, fences or road markings often represent a significant challenge for place recognition. Repeated structures are notoriously hard for establishing correspondences using multi-view geometry. They violate the feature independence assumed in the bag-of-visual-words representation which often leads to over-counting evidence and significant degradation of retrieval performance. In this work we show that repeated structures are not a nuisance but, when appropriately represented, they form an important distinguishing feature for many places. We describe a representation of repeated structures suitable for scalable retrieval and geometric verification. The retrieval is based on robust detection of repeated image structures and a suitable modification of weights in the bag-of-visual-word model. We also demonstrate that the explicit detection of repeated patterns is beneficial for robust visual word matching for geometric verification. Place recognition results are shown on datasets of street-level imagery from Pittsburgh and San Francisco demonstrating significant gains in recognition performance compared to the standard bag-of-visual-words baseline as well as the more recently proposed burstiness weighting and Fisher vector encoding.

Index Terms

Place Recognition, Bag of Visual Words, Geometric Verification, Image Retrieval.

- *A. Torii is with the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan.*
  E-mail: torii@ctrl.titech.ac.jp
- *J. Sivic is with the Inria, WILLOW, Departement d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.*
  E-mail: Josef.Sivic@ens.fr
- *M. Okutomi is with the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan.*
  E-mail: mxo@ctrl.titech.ac.jp
- *T. Pajdla is with the Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague.*
  E-mail: pajdla@cmp.felk.cvut.cz

Visual Place Recognition with Repetitive Structures

1 Introduction

Given a query image of a particular street or a building, we seek to find one or more images in the geotagged database *depicting the same place*. The ability to visually recognize a place depicted in an image has a range of potential applications including automatic registration of images taken by a mobile phone for augmented reality applications [1] and accurate visual localization for robotics [2]. Scalable place recognition methods [2], [3], [4], [5], [6] often build on the efficient bag-of-visual-words representation developed for object and image retrieval [7], [8], [9], [10], [11], [12]. In an offline pre-processing stage, local invariant descriptors are extracted from each image in the database and quantized into a pre-computed vocabulary of visual words. Each image is represented by a sparse (weighted) frequency vector of visual words, which can be stored in an efficient inverted file indexing structure. At query time, after the visual words are extracted from the query image, the retrieval proceeds in two steps. First a short-list of top ranked candidate images is obtained from the database using the bag-of-visual-words representation. Then, in the second verification stage, candidates are re-ranked based on the spatial layout of visual words.

A number of extensions of this basic architecture have been proposed. Examples include: (i) learning better visual vocabularies [13], [14]; (ii) developing quantization methods less prone to quantization errors [15], [16], [17]; (iii) combining returns from multiple query images depicting the same scene [7], [18]; (iv) exploiting the 3D or graph structure of the database [19], [20], [21], [22], [23], [24]; or (v) indexing on spatial relations between visual words [25], [26], [27].

In this work we develop a scalable representation for large-scale matching of repeated structures. While repeated structures often occur in man-made environments – examples include building facades, fences, or road markings – they are usually treated as nuisance and down-weighted at the indexing stage [4], [8], [28], [29]. In contrast, we develop a simple but efficient representation of repeated structures and demonstrate its benefits for place recognition in urban environments. In detail, we first robustly detect repeated structures in images by finding spatially localized groups of visual words with similar appearance. Next, we modify the weights of the detected repeated visual words in the bag-of-visual-word model, where multiple occurrences of repeated elements in the same image provide a *natural soft-assignment* of features to visual words. In addition, the contribution of repetitive structures is controlled to prevent dominating the matching scores. This is illustrated in Figure 1. Finally, we develop a geometric verification method that takes into account the detected repetitions and suppresses ambiguous tentative correspondences between repeated image patterns.

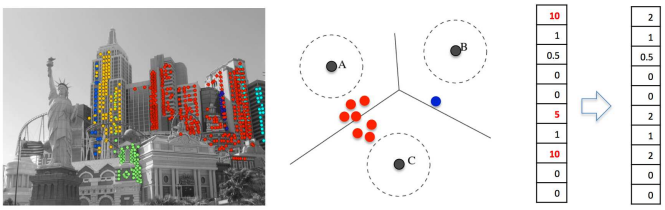


Fig. 1. **Overview of visual place recognition with repetitive structures.** Left: We detect groups of repeated local features (overlaid in colors). Middle: Repetitive features (shown in red) implicitly provide soft-assignment to multiple visual words (here A and C). Right: Truncating large weights (shown in red) in the bag-of-visual-word vectors prevents repetitions from dominating the matching score.

The paper is organized as follows. After describing related work on finding and matching repeated structures (Section 1.1), we review in detail (Section 2) the common tf-idf visual word weighting scheme and its extensions to soft-assignment [16] and repeated structure suppression [8]. In Section 3, we describe our method for detecting repeated visual words in images. In Section 4, we describe the proposed model for scalable matching of repeated structures, and in Section 5, we detail the proposed geometric matching that takes into account the detected repetitions. Experiments demonstrating the benefits of the developed representations are given in Section 6.

1.1 Related work

Detecting repeated patterns in images is a well-studied problem. Repetitions are often detected based on an assumption of a single pattern repeated on a 2D (deformed) lattice [30], [31], [32]. Special attention has been paid to detecting planar patterns [33], [34], [35] and in particular building facades [3], [36], [37], for which highly specialized grammar models, learnt from labelled data, were developed [38], [39].

Detecting planar repeated patterns can be useful for single view facade rectification [3] or even single-view 3D reconstruction [40]. However, the local ambiguity of repeated patterns often presents a significant challenge for geometric image matching [34], [41] and image retrieval [8].

Schindler *et al.* [34] detect repeated patterns on building facades and then use the rectified repetition elements together with the spatial layout of the repetition grid to estimate the camera pose of a query image, given a database of building facades. Results are reported on a dataset of 5 query images and 9 building facades. In a similar spirit, Doubek *et al.* [42] detect the repeated patterns in each image and represent the pattern using a single shift-invariant descriptor of the repeated element together with a simple descriptor of the 2D spatial layout. Their matching method is not scalable as they have to exhaustively compare repeated patterns in all images. In scalable image retrieval, Jegou *et al* [8] observe that repeated structures violate the feature independence assumption in the bag-of-visual-word model and test several schemes for down-weighting the influence of repeated patterns.

This paper is an extended version of [43] with detailed description of the proposed algorithms, new geometric verification method that takes into account the detected repetitive structures (Section 5), and additional experiments.

2 Review of visual word weighting strategies

In this section we first review the basic tf-idf weighting scheme proposed in text retrieval [44] and also commonly used for the bag-of-visual-words retrieval and place recognition [3], [4], [7], [8], [10], [11], [12], [26]. Then, we discuss the soft-assignment weighting [16] to reduce quantization errors and the "burstiness" model proposed by Jegou *et al.* [8], which explicitly down-weights repeated visual words in an image.

2.1 Term frequency–inverse document frequency weighting

The standard "term frequency–inverse document frequency" (*tf–idf*) weighting [44], is computed as follows. Suppose there is a vocabulary of M visual words, then each image is represented by a vector

$$\mathbf{y}_d = (y_1, ..., y_t, ..., y_M)^\top \tag{1}$$

of weighted visual word frequencies with components

$$y_t = \frac{n_{td}}{n_d} \log \frac{N}{N_t}, \tag{2}$$

where n_{td} is the number of occurrences of visual word t in image d, n_d is the total number of visual words in the image d, N_t is the number of images containing term t, and N is the number of images in the whole database. The weighting is a product of two terms: the *visual word frequency*, n_{td}/n_d, and the *inverse document (image) frequency*, $\log(N/N_t)$. The word frequency weights words occurring more often in a particular image higher (compared to visual word present/absent), whilst the inverse document frequency down-weights visual words that appear often in the database, and therefore do not help to discriminate between different images. The generalization of the idf weighting has been recently proposed by Zheng *et al* [45].

At the retrieval stage, images are ranked by the normalized scalar product (cosine of angle)

$$\text{score}_d = \frac{\mathbf{y}_q{}^\top \mathbf{y}_d}{\|\mathbf{y}_q\|_2 \ \|\mathbf{y}_d\|_2} \tag{3}$$

between the query vector $\mathbf{y}_q$ and all image vectors $\mathbf{y}_d$ in the database, where $\|\mathbf{y}\|_2 = \sqrt{\mathbf{y}^\top \mathbf{y}}$ is the L_2 norm of $\mathbf{y}$. The scalar product in Equation (3) can be implemented efficiently using inverted file indexing schemes.

2.2 Soft-assignment weighting

Visual words generated through descriptor clustering often suffer from quantization errors, where local feature descriptors that should be matched but lie close to the Voronoi boundary are incorrectly assigned to different visual words. To overcome this issue, Philbin *et al.* [16] soft-assign each descriptor to several (typically 3) closest cluster centers with weights set according to $\exp -\frac{d^2}{2\sigma^2}$, where d is the Euclidean distance of the descriptor from the cluster center and σ is a parameter of the method.

2.3 Burstiness weighting

Jegou *et al.* [8] studied the effect of visual "burstiness", i.e. that a visual-word is much more likely to appear in an image, if it has appeared in the image already. Burstiness has been also studied for words in text [46]. Jegou *et al.* observe by counting visual word occurrences in a large corpus of 1M images that visual words occurring multiple times in an image (e.g. on repeated structures) violate the assumption that visual word occurrences in an image are independent. Further, they observe that the bursted visual words can negatively affect retrieval results. The intuition is that the contribution of visual words with a high number of occurrences towards the scalar product in Equation (3) is too high. In the voting interpretation of the bag-of-visual-words model [26], bursted visual words vote multiple times for the same image. To see this, consider an example where a particular visual word occurs twice in a query and five times in a database image. Ignoring the normalization of the visual word vectors for simplicity, multiplying the number of occurrences as in (3) would result in 10 votes, whereas in practice only up to two matches (correspondences) can exist.

To address this problem Jegou *et al.* proposed to down-weight the contribution of visual words occurring multiple times in an image, which is referred to as intra-image burstiness. They experimented with different weighting strategies and empirically observed that down-weighting repeated visual words by multiplying the term frequency in Equation (2) by factor $\frac{1}{\sqrt{n_{td}}}$, where n_{td} is the number of occurrences, performed best. Similar

strategies to discount repeated structures when matching images were also used in [28], [29].

Note that Jegou *et al.* also considered a more precise description of local invariant regions quantized into visual words using an additional binary signature (Hamming embedding) [26] more precisely localizing the descriptor in the visual word Voronoi cell. The Hamming embedding is complementary to the method developed in this paper.

Contrary to down-weighting repeated structures based on globally counting feature repetitions across the entire image, we (i) explicitly detect localized image areas with repetitive structures, and (ii) use the detected local repetitions to adaptively adjust the visual word weights in the soft-assigned bag-of-visual words model. The two steps are described next.

3 Detection of Repetitive Structures

The goal is to segment local invariant features detected in an image into localized groups of repetitive patterns and a layer of non-repeated features (see Figure 2). Examples include detecting repeated patterns of windows on different building facades, fences, road markings or trees in an image (see Figure 3). We will operate directly on the extracted local features (rather than using specially designed features [36]) as the detected groups will be used to adjust feature weights in the bag-of-visual-words model for efficient indexing. The feature segmentation problem is posed as finding connected components in a graph.

In detail, we build an (undirected) feature graph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ with N vertices $\mathcal{F} = \{f_i\}_{i=1}^{N}$ consisting of local invariant features at locations $\mathbf{x}_i$, scales s_i and with corresponding SIFT descriptors $\mathbf{d}_i$. Each SIFT descriptor is further assigned to the K nearest visual words $\mathcal{W}_i^K = \{w_i^k\}_{k=1}^K$ in a pre-computed visual vocabulary (see Section 6 for details). Two vertices (features) f_i and f_j are connected by an edge if they have close-by image positions as well as similar scale and appearance. More formally, a pair of vertices f_i and f_j is connected by an edge if the following three conditions are satisfied:

1) The spatial L_2 distance $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ between features satisfies $\|\mathbf{x}_i - \mathbf{x}_j\|_2 < \gamma\,(s_i + s_j)$ where γ is a constant (we set $\gamma = 10$ throughout all our experiments);
2) The ratio σ of scales of the two features is in $0.5 < \sigma < 2$;
3) The features share at least one common visual word in their top K visual word assignments, where K is typically 50. Note that this condition avoids directly thresholding the distance between the SIFT descriptors of the two features, which we found unreliable.

Having built the graph, we group the vertices (image features) into disjoint groups by finding connected components of the graph [47]. These connected components group together features that are spatially close, and are also similar in appearance as well as in scale. In the following, we will call the detected feature groups "repttiles" for "tiles (regions) of repetitive features". The repttile detection is summarized in Algorithm 1.

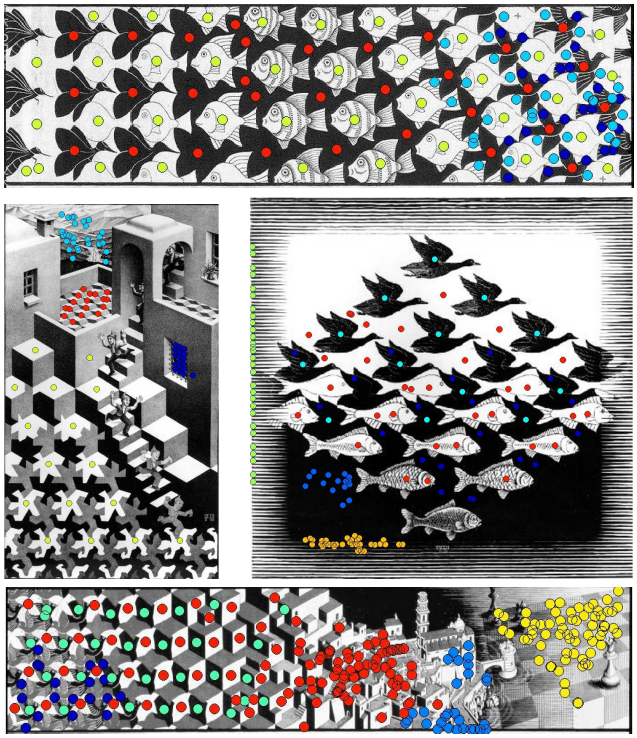


Fig. 2. Examples of detected repetitive patterns of local invariant features ("repttiles"). The different repetitive patterns detected in each image are shown in different colors. The detection is robust against local deformation of the repeated element and makes only weak assumptions on the spatial structure of the repetition.

Algorithm 1 Repttile detection

Input $\mathcal{C}$: Visual word centroids.
$\mathcal{F}$: N vertices (features) consisting of
$\mathbf{x}_i$: location, s_i: scale, $\mathbf{d}_i$: descriptor.
Output $\mathcal{E}$: Edges of undirected graph $\mathcal{G}$.

Initialize $e_{ij} \in \mathcal{E} := \emptyset$.
for $i = 1, \ldots, N$ **do**
  For descriptor $\mathbf{d}_i$ find the K nearest visual words $\mathcal{W}_i^K = \{w_i^k\}_{k=1}^K$ from vocabulary $\mathcal{C}$.
for $i = 1, \ldots, N$ **do**
  For feature f_i retrieve matching features f_j s.t. $j \in \mathcal{J}$: $\mathcal{W}_i^K \cap \mathcal{W}_j^K \neq \emptyset$.
  for $j \in \mathcal{J}$ **do**
    if e_{ij} = `FALSE` **then**
      if $0.5 < s_i/s_j < 2$ **then**
        if $\|\mathbf{x}_i - \mathbf{x}_j\| < \gamma\,(s_i + s_j)$ **then**
          $e_{ij} = e_{ji}$:= `TRUE`. %Create edge.

Figures 2 and 3 show a variety of examples of detected patterns of repeated features. Only connected components with more than 20 image features are shown as colored dots. Note that the proposed method makes

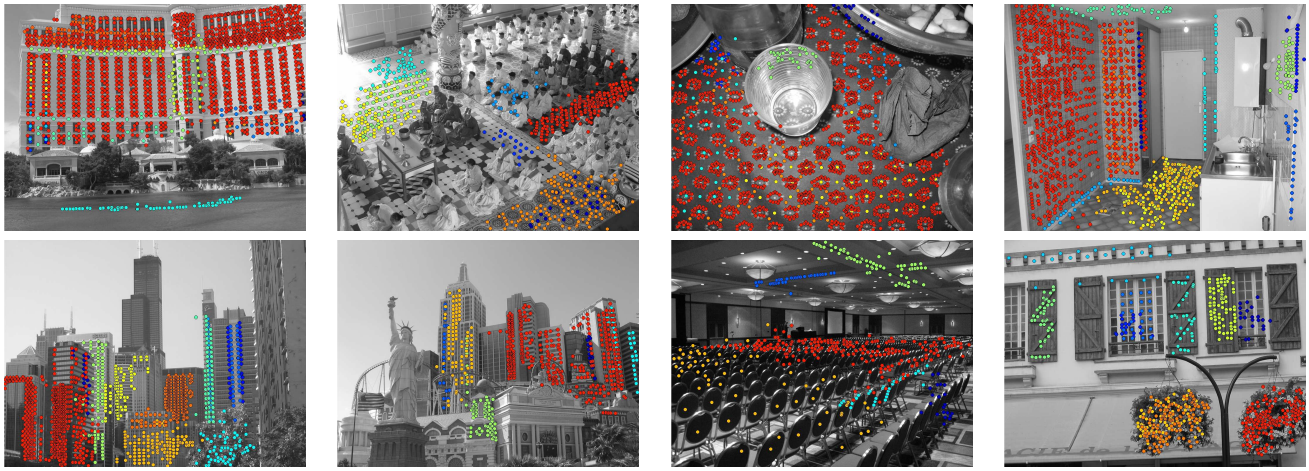


Fig. 3. Examples of detected repetitive patterns of local invariant features ("repttiles") in images from the INRIA Holidays dataset [8]. The different repetitive patterns detected in each image are shown in different colors. The color indicates the number of features in each group (red indicates large and blue indicates small groups). Note the variety of detected repetitive structures such as different building facades, trees, indoor objects, window tiles or floor patterns.

only weak assumptions on the type and spatial structure of repetitions, not requiring or attempting to detect, for example, feature symmetry or an underlying spatial lattice.

4 Representing Repetitive Structures for Scalable Retrieval

In this section we describe our image representation for efficient indexing taking into account the repetitive patterns. The proposed representation is built on two ideas. First, we aim at representing *the presence* of a repetition, rather than measuring the actual number of matching repeated elements. Second, we note that different occurrences of the same visual element (such as a facade window) are often quantized to different visual words because of the noise in the description and quantization process as well as other non-modeled effects such as complex illumination (shadows) or perspective deformation. We take the advantage of this fact and design a descriptor quantization procedure that *adaptively soft-assigns* local features with more repetitions in the image to fewer nearest cluster centers. The intuition is that the multiple examples of a repeated feature provide a natural and accurate soft-assignment to multiple visual words.

Formally, an image is represented by a bag-of-visual-words vector

$$\mathbf{z}_d = (z_1, ..., z_t, ..., z_M)^\top \tag{4}$$

where the t-th visual word weight

$$z_t = \begin{cases} r_t & \text{if } 0 \leq r_t < T \\ T & \text{if } T \leq r_t \end{cases} \tag{5}$$

is obtained by thresholding weights r_t by a threshold T. Note that the weighting described in Equation (5) is similar to burstiness weighting, which down-weights repeating visual words. Here, however, we represent highly weighted (repeating) visual words with a constant T as the goal is to represent the occurrence (presence/absence) of the visual word, rather than measuring the actual number of occurrences (matches).

Weight r_t of the t-th visual word in image d is obtained by aggregating weights from adaptively soft-assigned features across the image taking into account the repeated image patterns. In particular, each feature $f_i \in \mathcal{F}_d$ detected in image d is assigned to the α_i nearest (in the feature space) visual words $\mathcal{W}_i^{\alpha_i} = \{w_i^k\}_{k=1}^{\alpha_i}$. Thus, w_i^k, for $1 \leq k \leq \alpha_i$, is the index of the k-th nearest visual word of f_i. The number α_i, which varies between 1 and $\alpha_{\max}$, will be defined below. Weight r_t is computed as

$$r_t = \sum_{i=1}^{N} \sum_{k=1}^{\alpha_i} 1[w_i^k = t] \frac{1}{2^{k-1}} \tag{6}$$

where the indicator function $1[w_i^k = t]$ is equal to 1 if visual word t is present at the k-th position in $\mathcal{W}_i^{\alpha_i}$. This means that weight r_t is obtained as the sum of contributions from all assignments of visual word t over all features in $\mathcal{F}_d$. The contribution of an individual assignment depends on the order k of the assignment in $\mathcal{W}_i^{\alpha_i}$ by the weight $1/(2^{k-1})$. The number α_i is computed by the following formula

$$\alpha_i = \left\lceil \alpha_{\max} \frac{\log(\frac{n_d+1}{m_i})}{\max_{i \in \mathcal{F}_d} \log(\frac{n_d+1}{m_i})} \right\rceil \tag{7}$$

where α_{max} is the maximum number (upper bound) of assignments ($\alpha_{max} = 3$ in all our experiments), m_i is the number of features in the repttile of which f_i belongs, and n_d is the total number of features in the image d. We use $\lceil x \rceil = ceiling(x)$, *i.e.* $\lceil x \rceil$ is the smallest integer greater than or equal to x. This adaptive soft-assignment is summarized in Algorithm 2. Note that image features belonging to relatively larger repttiles are soft-assigned

Algorithm 2 BoVW weighting with adaptive assignment

Input

$\mathcal{W}_i^K$: K-nearest visual words for each feature i in $\mathcal{F}$.

Output

r_t: Weights of the bag-of-visual-word vector.

1: Initialize $r_t := 0$ for $t = 1, \ldots, M$.
2: **for** $i = 1, \ldots, N$ **do**
3: Compute number of assignments α_i by Eq. (7).
4: **for** $k = 1, \ldots, \alpha_i$ **do**
5: $p := w_i^k \in \mathcal{W}_i^K$. %Retrieve the k-th visual word.
6: $r_p := r_p + 1/2^{k-1}$.
7: Apply thresholding in Eq. 5 to r_t for $t = 1, \ldots, M$.

to fewer visual words as image repetitions provide a natural soft-assignment of the particular repeating scene element to multiple visual words (see Figures 4 and 5). This adaptive soft-assignment is more precise and less ambiguous than the standard soft-assignment to multiple nearest visual words [16] as will be demonstrated in Section 6.

5 Geometric Verification with Repttile Detection

In this section we describe our geometric verification method that takes advantage of the detected repeated patterns. Similarly to the standard image retrieval, the place recognition accuracy can be significantly improved by re-ranking retrieved images based on geometric consistency of local features [11]. The retrieved images in the initial shortlist are typically re-ranked using the number of inliers that are consistent with the 2D affine geometric transformation, homography, or the epipolar constraint.

The geometric verification has two steps: (i) generating tentative matches based on local feature appearance, and (ii) finding subsets of matches consistent with the geometric transformation. The tentative matches are typically found by matching raw (SIFT) feature descriptors [48] or via visual word matching [11], [12]. For large scale matching problems, visual word matching is preferred due to its memory efficiency as we can store only 4 bytes for an unsigned-int32 visual word ID instead of 128 bytes for an unsigned-int8 SIFT descriptor. However, as reported in [11], [49], visual word matching usually results in some drop in matching accuracy due to quantization effects. In addition, when using visual words it is not straightforward to find the most distinctive matches as can be done with raw descriptors using Lowe's first to second nearest neighbor ratio test [48]. We develop a method for generating tentative matches for geometric verification that addresses both these issues.

First, to overcome the quantization effects we assign descriptors to multiple visual words (multiple assignment). This is similar to [16] but to minimize false matches we take advantage of the detected repeated patterns and only compute tentative matches from the

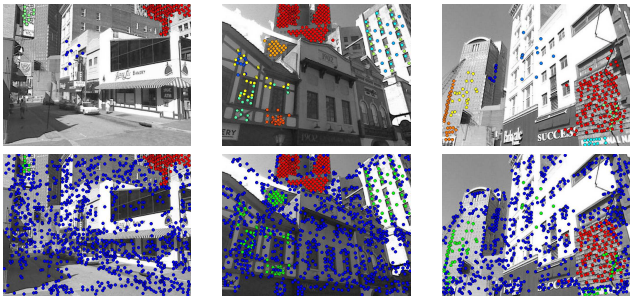


Fig. 4. Examples of adaptive soft-assignment with "repttile" detection in images from the Pittsburgh dataset. (Top) The repttiles composed from more than 20 image features are shown in different colors (red indicates large and blue indicates small groups, similarly to Figure 3). (Bottom) The number of visual word assignments of each feature is adaptively defined by the number of features in the repttile as in Equation (7). The color indicates the number of multiple assignments, red $= 1$, green $= 2$ and blue$= 3$. Features belonging to larger repttiles are assigned to fewer visual words (red) but discriminative features (blue and green) are assigned to multiple visual words (up to 3).

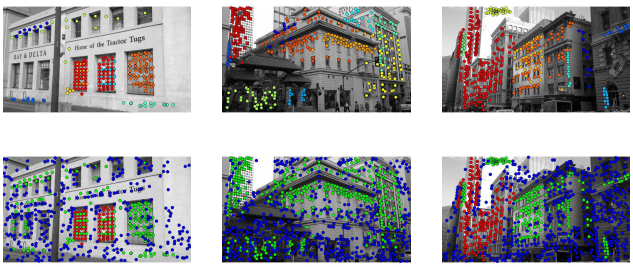


Fig. 5. Examples of adaptive soft-assignment with "repttile" detection in images from the San Francisco dataset. See the caption of Figure 4 for details.

distinct visual words with the limited amount of repetitions in the image. This is implemented by removing features from the largest repttiles (those where $\alpha_i = 1$, see Section 4). The remaining features are assigned to multiple (K) nearest visual words, where K is typically 50. The multiple assignment is performed only on the query side, and therefore, it is not necessary to store K visual words for all database images as in [16]. Note that no additional computations are required as the K nearest visual words are already computed during the repttile detection (Section 3). The results in Section 6 demonstrate that it is beneficial to remove the heavily repeated features during geometric verification as they are highly ambiguous for the task of establishing feature to feature correspondences. The visual word matching with repttile removal and multiple assignments is summarized in Algorithm 3.

Second, we develop a variant of Lowe's first to second nearest neighbor ratio test [48] for visual word matching

Algorithm 3 Visual word matching with repttile removal and multiple assignments

Input
$\mathcal{W}_{iq}^K$: Visual words of $\mathcal{F}_q$ in query image I_q.
α_{iq}: Numbers of adaptive assignments for $\mathcal{F}_q$.
$\mathcal{W}_{id}^1$: Visual words of $\mathcal{F}_d$ in image I_d.
α_{id}: Numbers of adaptive assignments for $\mathcal{F}_d$.
Output
m_i: Indices of matches to the query features.

Initialize $m_i := \emptyset$ for $i = 1, \ldots, N_q$.
Remove $f_{id} \in \mathcal{F}_d$ if $\alpha_{id} = 1$.
%Remove large repttiles in I_d.
for $i = 1, \ldots, N_q$ **do**
  if $\alpha_{iq} > 1$ **then** %Use only small repttiles in I_q.
    $k := 0$
    while $k < K$ **do**
      $k := k + 1$
      Seek a set $\mathcal{J}$ where $\{j \in \mathcal{J} \mid w_{iq}^k \cap w_{jd}^1 \neq \emptyset\}$.
      %Find visual word match.
      if $\mathcal{J} \notin \emptyset$ **then**
        Pick an index $j \in \mathcal{J}$ randomly.
        $m_i := j$
        break

when the raw (SIFT) descriptors for the images in the database are not available. The main idea of the original ratio test [48] is to compute, for each query image descriptor $\mathbf{d}_{iq}$, the ratio

$$L = \frac{\|\mathbf{d}_{iq} - \mathbf{d}_{NN_1}\|_2}{\|\mathbf{d}_{iq} - \mathbf{d}_{NN_2}\|_2}, \tag{8}$$

where $\mathbf{d}_{NN_1}$ is the first nearest neighbor and $\mathbf{d}_{NN_2}$ is the second nearest neighbor of the query descriptor $\mathbf{d}_{iq}$ in the candidate database image. The ratio L lies between 0 and 1. The ratio is close to 1 if the query feature $\mathbf{d}_{iq}$ is non-distinctive. While this test works extremely well in practice, it requires storing (or re-computing) local feature descriptors for both the query image and all database images, which, as argued above, is prohibitive for large databases.

To address this issue, we develop an asymmetric version of the ratio test, which is inspired by the asymmetric distance computation in product quantization [15], and which only requires knowing the local feature descriptors for the query image. The local feature descriptors in the database images are represented by their quantized representation, i.e. the centroids of their closest visual words. In detail, we replace each feature descriptor $\mathbf{d}_{id}$ in database image d by its quantized descriptor corresponding to the centroid of its closest visual word $\mathbf{c}_t \in \mathcal{C}$. The approximate ratio is then computed as

$$L_a = \frac{\|\mathbf{d}_{iq} - \mathbf{c}_{NC_1}\|_2}{\|\mathbf{d}_{iq} - \mathbf{c}_{NC_2}\|_2}, \tag{9}$$

where $\mathbf{d}_{iq}$ is the descriptor of the local feature in the query image, $\mathbf{c}_{NC_1}$ is its closest (quantized) descriptor in the database image and $\mathbf{c}_{NC_2}$ is its second closest (quantized) descriptor in the database image. L_a also lies between 0 and 1 and is close to 1 if the descriptor match is ambiguous. The intuition is that NC_1 and NC_2 are approximations of the first and second nearest neighbors NN_1 and NN_2 in the original ratio test given by Equation (8). Note that the asymmetric ratio (9) can be computed efficiently as the distances $\|\mathbf{d}_{iq} - \mathbf{c}_{NC}\|_2$ between the query descriptor and multiple (up to 50) closest cluster centers are pre-computed during the assignment of query descriptors to visual words. In addition, NC_1 and NC_2 can be efficiently found by only accessing the inverted file list.

Results demonstrating benefits of both tentative matching with reptile detection and the asymmetric ratio test are shown in Section 6.

6 Experiments

In this section we describe the experimental validation of our approach. First, we describe the experimental set-up (Section 6.1), describe the evaluation metric (Section 6.2) and give the implementation details (Section 6.3). Next, we evaluate place recognition performance of the proposed representation with repttile detection and adaptive soft-assignment and compare it with the state-of-the-art BoVW baseline methods (Section 6.4). Then we evaluate the sensitivity to the different method parameters (Section 6.5) and compare performance with compact image descriptors (Section 6.6). Finally, we experimentally evaluate the proposed geometric verification method (Section 6.7).

6.1 Datasets

We perform experiments on the following two geotagged image databases.

6.1.1 Pittsburgh dataset

The Pittsburgh dataset is formed by $254,064$ perspective images generated from $10,586$ Google Street View panoramas of the Pittsburgh area downloaded from the the Internet. From each panorama of $6,656 \times 3,328$ pixels, we generate 24 perspective images of 640×480 pixels (corresponding to 60 degrees of horizontal FoV) with two pitch directions $[4°, 26.5°]$ and 12 yaw $[0°, 30°, ..., 360°]$ directions. This is a similar setup to [3]. As testing query images, we use $24,000$ perspective images generated from $1,000$ panoramas randomly selected from $8,999$ panoramas of the Google Pittsburgh Research Data Set[1]. The datasets are visualized on a map in Figure 6(a). This is a very challenging place recognition set-up as the query images were captured in a different session than the database images and depict the same places from different viewpoints, under very different illumination conditions and, in some cases, in different seasons. Note also the high number of test query images compared to other existing datasets [3], [4].

1. Provided and copyrighted by Google.

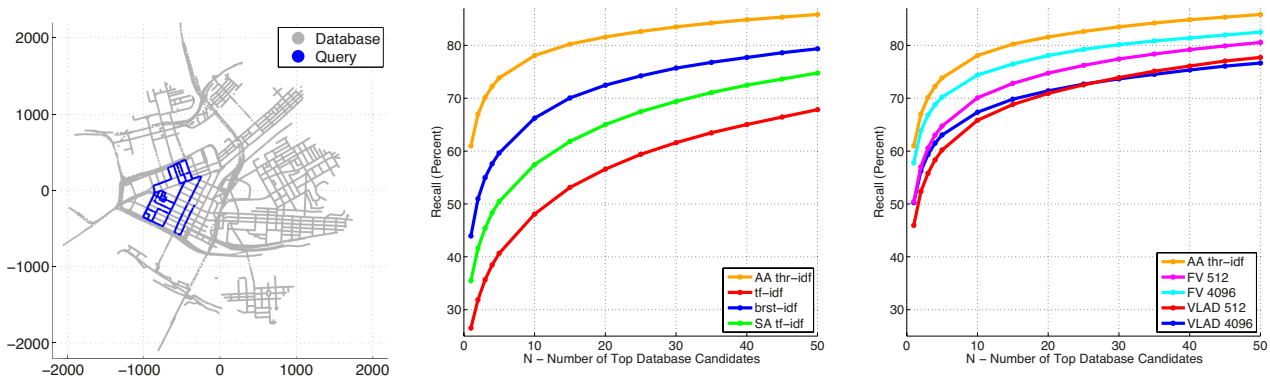


Fig. 6. **Evaluation on the Pittsburgh dataset.** (a) Locations of query (blue dots) and database (gray dots) images. (b-c) The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for the proposed method (AA thr-idf) compared to several other methods.

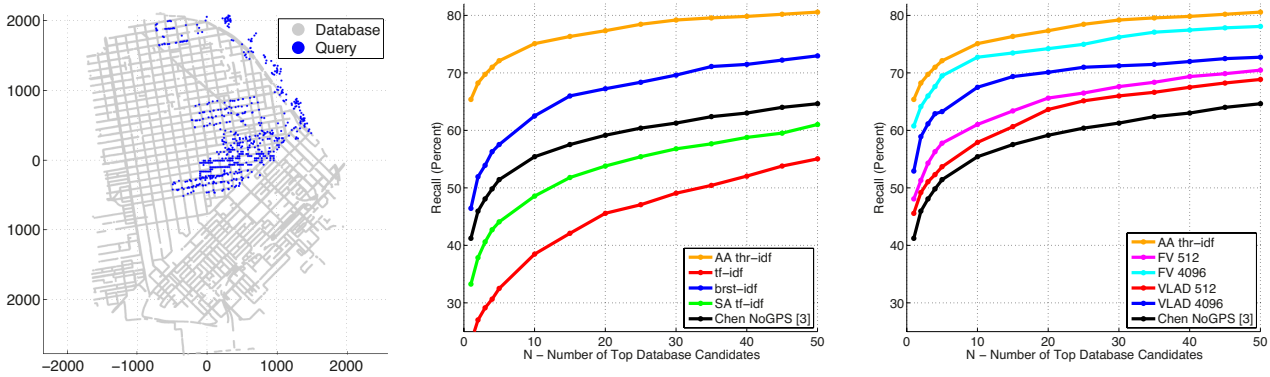


Fig. 7. **Evaluation on the San Francisco dataset.** (a) Locations of query (blue dots) and database (gray dots) images. (b-c) The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for the proposed method (AA thr-idf) compared to several other methods.

The ground truth is derived from the (known) GPS positions of the query images. We have observed that GPS positions of Street View panoramas are often snapped to the middle of the street. The accuracy of the GPS positions hence seems to be somewhere between 7 and 15 meters.

6.1.2 San Francisco visual place recognition benchmark

This dataset consists of the geotagged images formed by $1,062,468$ perspective central images (PCIs), $638,090$ perspective frontal images (PFIs), and 803 cell phone images. We use the PCIs as the geotagged image database and the cell phone images as testing query images. The dataset is visualized on a map in Figure 7(a). This dataset involves challenges similar to the Pittsburgh dataset. Some query images undergo more severe viewpoint changes than those in the Pittsburgh dataset because the cell phone query images are captured on the street-side whereas the PCI database images are captured from the middle of street.

Each image in this dataset has labels of visible buildings computed using 3D building models [50]. The ground truth is given by matching the building IDs between query and database images[2].

6.2 Evaluation metric

Similarly to [3], [4], [51], we measure the place recognition performance by the fraction of correctly recognized queries. Following [3], we denote the fraction of correctly recognized queries as "recall"[3]. We measure performance by recall on both datasets. However, the definition of a correctly recognized query is different between the two datasets due to different type of available ground truth annotations. For the Pittsburgh dataset, a query is deemed correctly recognized if at least one of the top N retrieved database images is within m meters from the ground truth position of the query. For the San Francisco dataset, the query is deemed

2. We note that 60 query images do not have the corresponding ground truth building label in the dataset.

3. While this terminology is slightly different from image retrieval, it meets the definition of recall, i.e. it measures the fraction of true positives (correctly recognized queries) over all positives (all query images).

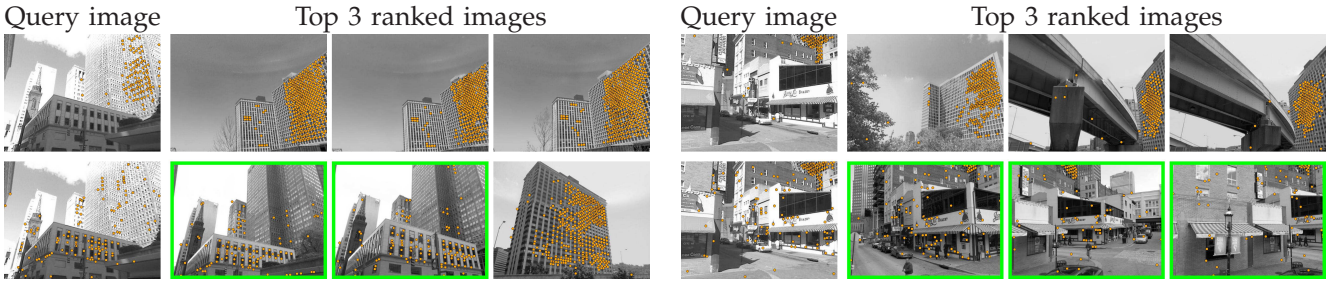


Fig. 8. **Examples of place recognition results on the Pittsburgh dataset.** Each figure shows the query image (left column) and the three best matching database images (2nd to 4th column) using the baseline burstiness method [8] (top row) and the proposed repttile detection and adaptive assignment (bottom row). The green borders indicate correctly recognized images. The orange dots show the visual word matches between the query image (1st column) and the best matching database image (2nd column). The visual word matches are also displayed on the 2nd and 3rd best matching images. Please note that for improved clarity, the matching visual words in the query are not shown for the second and third match.

f5a

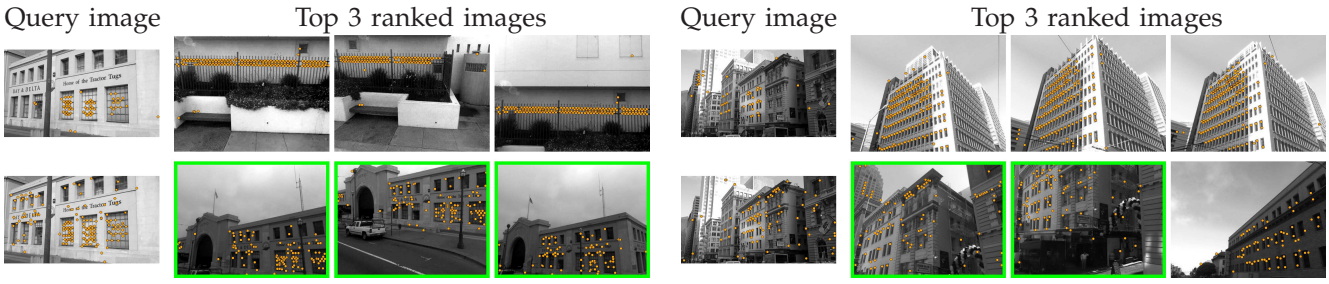


Fig. 9. **Examples of place recognition results on the San Francisco dataset.** See the caption of Figure 8 for details.

correctly recognized if at least one of the top N retrieved database images has the ground truth building ID match with the query. We evaluate the fraction of correctly recognized queries (recall) for the different lengths N of the candidate shortlist.

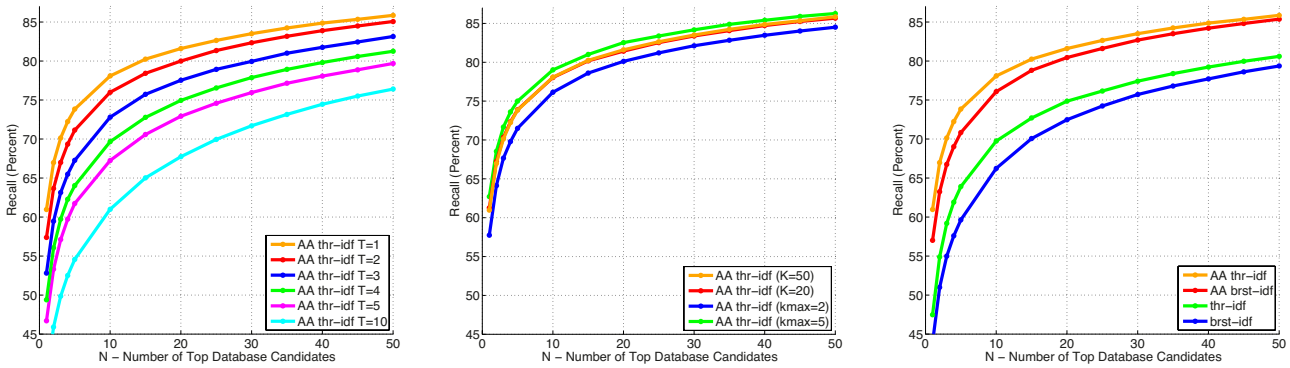
6.3 Implementation details

We build a visual vocabulary of 200,000 visual words by approximate k-means clustering [11], [52]. We have not observed any significant improvements by using larger vocabularies. The vocabulary is built from features detected in a subset of $100,000$ randomly selected database images in each dataset. We use the upright SIFT descriptors [48], [53] for each feature (assuming the upright image gravity vector) followed by the RootSIFT normalization [54], *i.e.* L1 normalization and square root weighting. We have not used the histogram equalization suggested by [3] as it did not improve results using our visual word setup.

6.4 Comparison with BoVW baselines

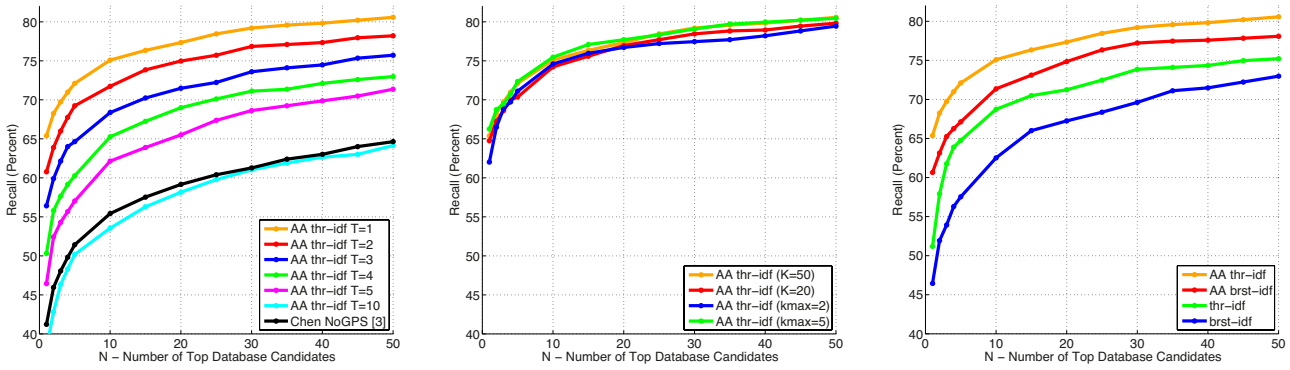
We compare results of the proposed adaptive soft-assignment approach (AA thr-idf) with several baselines: the standard tf-idf weighting (tf-idf) [11], burstiness weights (brst-idf) [8], and the standard soft-assignment weights [16] (SA tf-idf). Results for the two datasets are summarized below.

- **Pittsburgh.** Results for different methods for $m = 25$ meters and varying value of N are shown in Figure 6 (b). Figure 8 shows examples of place recognition results. Notice that the top ranked results by the baseline burstiness method [8] still suffer from repetitive structures that dominate visual word matching (Figure 8 (top row)). However, the proposed repttile detection and adaptive soft-assignment effectively down-weight repetitive structures, while mitigating "quantization effects" for the distinctive features (Figure 8 (bottom row)).
- **San Francisco.** Results for the different methods are shown in Figure 7. The results of [3] were obtained directly from the authors but to remove the effect of geometric verification we ignored the threshold on the minimum number of inliers by setting $T_{PCI} = 0$. Note also that the GPS position of the query image was not used for any of the compared methods. The pattern of results is similar to the Pittsburgh data with our adaptive soft-assignment method (AA thr-idf) performing best and significantly better than the method of [3] underlying the importance of handling repetitive structures for place recognition in urban environments. Example place recognition



(a) Threshold T (b) K and $\alpha_{\max}$ (c) AA and T

Fig. 10. **Sensitivity to different parameters on the Pittsburgh dataset.** The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for different parameter setup.



(a) Threshold T (b) K and $\alpha_{\max}$ (c) AA and T

Fig. 11. **Sensitivity to different parameters on the San Francisco dataset.** The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for different parameter setup.

results demonstrating benefits of the proposed approach are shown in Figure 9.

6.5 Sensitivity to parameters

Here we list the main parameters of our method and evaluate its sensitivity to their settings.

- **The weight threshold T.** The weight threshold T in Equation (5) is an important parameter of the method and its setting may depend on the dataset and the size of the visual vocabulary. Since 97% of visual word weights r_t (see Equation (6)) are ≤ 1 (measured on the Pittsburgh database), setting $T = 1$ effectively down-weights the bursted visual words. Figures 10(a) and 11(a) show the evaluation of place recognition performance for different values of T. We use $T = 1$ (unless stated otherwise).
- **The number of visual word assignments K.** Figures 10(b) and 11(b) show the method is fairly insensitive to the choice of the number K of visual word assignments for repttile detection, where values of 20 (AA thr-idf ($K = 20$)) and the standard 50 (AA thr-idf ($K = 50$)) result in a similar performance. We use $K = 50$ (unless stated otherwise).
- **The maximum repttile soft-assignment $\alpha_{\max}$.** We have also tested different values of the adaptive soft-assignment parameter $\alpha_{\max}$ (Equations (6) and (7)). Figures 10(b) and 11(b) again show the method is fairly insensitive to its choice, where values of 2 (AA thr-idf (amax=2)), 3 (AA thr-idf ($K = 50$)), and 5 (AA thr-idf (amax=5)) result in a similar performance. We use $\alpha_{max} = 3$ following [16] (unless stated otherwise). Note that the base of the exponential in Equation (6) is chosen so that the weights decrease with increasing k and we found $1/2$ to work well. In general, this value needs to be set experimentally, similarly to the sigma parameter in the standard soft-assignment [16].
- **Which stage helps.** Here, we evaluate separately the benefits of the two components of the proposed method (weight thresholding and adaptive soft-assignment) compared to the baseline burstiness weights. Figures 10(c) and 11(c) show separately results for (i) thresholding using Equation (5) (thr-idf) and (ii) adaptive soft-assignment using Equations (6) and (7) (AA brst-idf). Combining the two components results in a further improvement (AA

TABLE 1
Scalability and place recognition performance on the Pittsburgh dataset.

Method	Memory (GB)	Recall top 1	top 5	top 50
AA thr-idf	3.17	60.97	73.83	85.85
tf-idf	1.21	26.52	40.66	67.85
brst-idf	1.21	43.93	59.64	79.36
SA tf-idf	3.54	35.46	50.45	74.77
FV 512	0.52	50.41	64.73	80.56
FV 4096	4.16	57.78	70.20	82.51

TABLE 2
Scalability and place recognition performance on the San Francisco dataset.

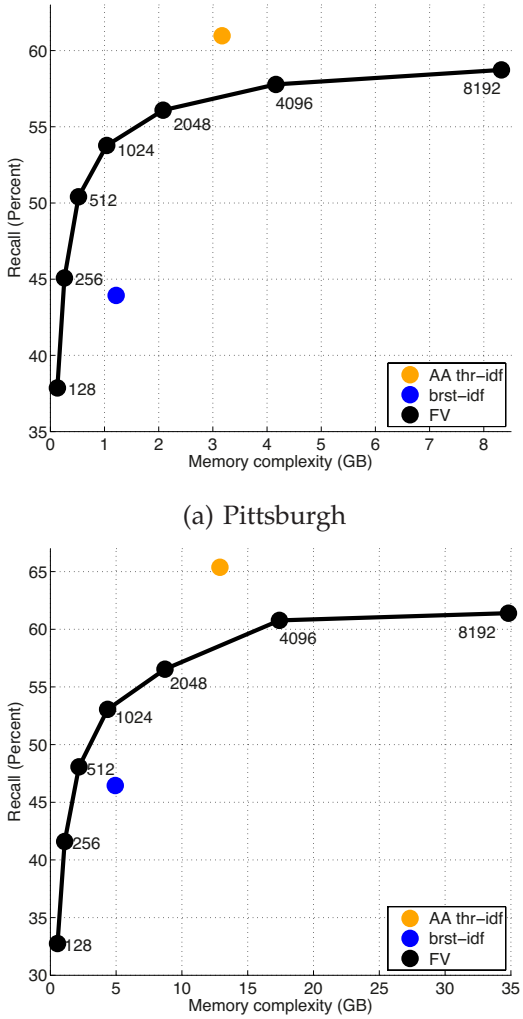
Method	Memory (GB)	Recall top 1	top 5	top 50
AA thr-idf	12.89	65.38	72.10	80.57
tf-idf	4.94	23.16	32.50	55.04
brst-idf	4.94	46.45	57.53	72.98
SA thr-idf	14.38	33.25	44.08	61.02
FV 512	2.18	48.07	57.78	70.49
FV 4096	17.41	60.77	69.49	78.08

thr-idf).

6.6 Comparison with compact descriptors

We compare results of the proposed adaptive soft-assignment approach with Vector of Locally Aggregated Descriptors (VLAD) and Fisher Vector (FV) matching [55]. Following [55], we construct VLAD and FV from RootSIFT descriptors reduced to 64 dimensions by PCA. The 512 centroids for VLAD and the 512 Gaussian mixture components for FV were trained on the same training images, which were used for the BoVW methods. As in [55], resulting 512×64 dimensional descriptors are then reduced to 512 (VLAD 512, FV 512) or $4,096$ (VLAD 4096, FV 4096) dimensions using PCA. The similarity between a query and database images is measured by the normalized scalar product. Figures 6(c) and 7(c) show the place recognition performance of our method compared to VLAD and FV.

Our adaptive soft-assignment can be indexed using standard inverted files and in terms of memory requirements compares favorably to the standard soft-assignment and Fisher vector representations. For the database of BoVW vectors, we count the memory complexity as 8 bytes per visual word entry (4 bytes for the index and 4 bytes for the weight). For the FV, we count 4 bytes (single precision) per dimension. Tables 1 and 2 show that the proposed BoVW representation (AA thr-idf) has a smaller memory footprint than soft-assignment [16] (SA tf-idf) and FV $4,096$ while achieving better place recognition performance. The memory complexity vs. recognition accuracy (recall) are plotted for the two datasets in Figure 12. For a given memory complexity, the proposed method (AA thr-idf) clearly outperforms the FV matching.



(a) Pittsburgh

(b) San Francisco

Fig. 12. **Memory complexity vs. place recognition performance.** The memory footprint of the database (x-axis) vs. the fraction of correctly recognized queries (Recall, y-axis) at the top 1 rank by the proposed method (AA thr-idf, orange dot), the baseline burstiness weighting (brst-idf, blue dot), and the Fisher vector representations (FV, black dots). The numbers beside the black dots indicate the dimension of each Fisher vector (reduced by PCA).

We note that the memory requirements can be further reduced for both our adaptive soft-assignment and the Fisher vector representations. Our adaptive weights (Equations (5) and (6)) can be mapped to a small set of integers when a small $\alpha_{\max}$ is used ($\alpha_{max} = 3$ in all our experiments). This reduces the memory requirements for storing our adaptive weights by a factor of 8. In addition, the inverted files can be compressed (without a loss of performance) which further reduces the required memory, typically by a factor of 4 [56]. When combined, the required memory for our adaptive soft-assignment can be reduced by a factor of 32 with no loss in place recognition performance. For Fisher

TABLE 3
Re-ranking with geometric verification on the Pittsburgh dataset.

Method	Multi-assign	Ratio test	Repttile removal	Recall (top 1)	Recall (top 3)	# of tent. matches	Computation time (sec)	Total memory (GB)
Initial ranking (AA thr-idf)				60.97	70.14	-	-	3.17
Raw descriptor		x		76.68	80.86	64.10	0.91	29.12
Visual word matching [11]				60.84	70.63	29.40	0.41	4.69
Visual word matching with the proposed method	x			66.93	75.68	252.99	4.94	4.69
	x	x		67.23	76.78	146.06	1.34	4.69
	x		x	**71.44**	**78.14**	108.40	1.03	4.88
	x	x	x	70.41	77.41	83.14	0.64	4.88

TABLE 4
Re-ranking with geometric verification on the San Francisco dataset.

Method	Multi-assign	Ratio test	Repttile removal	Recall (top 1)	Recall (top 3)	# of tent. matches	Computation time (sec)	Total memory (GB)
Initial ranking (AA thr-idf)				65.38	69.74	-	-	12.89
Raw descriptor		x		76.09	78.58	69.99	2.77	104.01
Visual word matching [11]				68.62	72.73	29.66	0.82	18.25
Visual word matching with the proposed method	x			73.47	77.09	227.72	8.29	18.25
	x	x		73.35	77.83	102.01	3.00	18.25
	x		x	**75.47**	77.83	134.75	2.06	18.92
	x	x	x	74.22	**77.96**	78.08	1.43	18.92

vectors, searching with product quantization [15], [55] in a standard setting ($m = 16$, $k' = 1,024$) typically reduces the memory requirements by a factor of 50 with a marginal loss in recognition performance. However, by inspecting Figures 12(a) and (b) we note that the Fisher vector performance consistently saturates for high dimensions. As a result, we do not expect the difference in the typical compression factors (32 for our method vs. 50 for FV) to affect the observed pattern of results.

6.7 Re-ranking by geometric verification

Here we evaluate the benefits of geometric verification. We re-rank the initial shortlist of top 50 images by the number of verified matches (inliers). We compare results of the proposed geometric verification method (Section 5) with standard baselines: the nearest neighbor matching using the raw descriptors with Lowe's ratio test [48] (threshold$= 0.9$) and visual word matching [11]. We evaluate separately the different components of the proposed method for obtaining tentative matches: (i) visual word matching with multiple assignments to different visual words (multi-assign), (ii) asymmetric ratio test (ratio test) (threshold$= 0.9$) and (iii) removing highly repetitive features (repttile removal). For the geometric verification of the input tentative matches, we use the LO-RANSAC [11], [57]. We also tested the standard RANSAC [58] and PROSAC [59] but observed no significant difference in performance.

Tables 3 and 4 show the percentage of correctly recognized queries (Recall) after geometric verification, the average number of tentative matches, and the average time to perform geometric verification for all the top 50 candidate images in the shortlist. The last column indicates the total memory footprint required to store the descriptors and other feature information to perform the geometric verification. We report recall at top 1 and top 3 as only the very top retrieved images are important for practical place recognition applications. On both datasets, the proposed visual word matching with multiple assignment combined with repttile removal and asymetric ratio test (last row in the table) significantly improves the recall compared to the standard visual word matching [11] while keeping similar memory footprint and slightly higher but still reasonable (under 1.5s) computation time.

Note that the reported computation time includes only the time to perform geometric verification using LO-RANSAC not the time to obtain the tentative matches. This explains the low reported timings for raw descriptor matching. In practice, raw descriptor matching is dominated by computing the tentative correspondences. In contrast, virtually no additional computations are required for visual word matching. Also note that the asymmetric ratio test does not always increase recall compared to repttile removal but further reduces the number of tentative matches and computation time.

Inspecting the memory footprint in more detail, we note that visual word matching [11] requires only the coordinates of keypoints, 4 bytes (single precision) $\times$ 2 (dimension), which amounts to 1.52 GB on the Pittsburgh and 5.36 GB on the San Francisco datasets, respectively. The raw descriptor matching in addition requires storing the raw SIFT descriptors (128×1 byte for each feature) that amounts, in total, to 24.43 GB and 85.76 GB for the two datasets, respectively. In comparison, the proposed multiple visual word matching with repttile removal requires only storing the repttile labels α_i (Equation (7), 1 byte for each feature) that amounts only to additional 0.19 GB and 0.67 GB compared to the standard visual word matching [11].

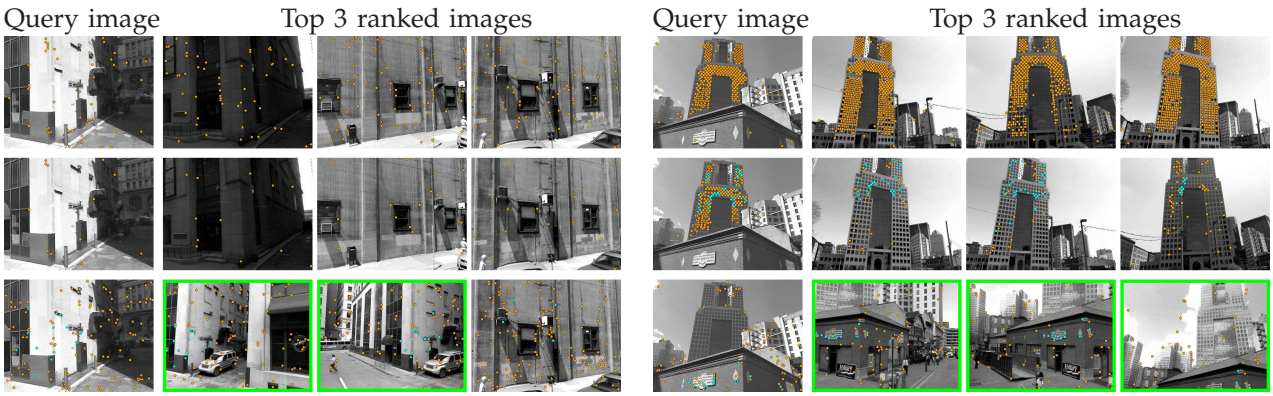


Fig. 13. **Examples of re-ranking by geometric verification on the Pittsburgh dataset.** Each figure shows the query image (left column) and the three best matching database images (2nd to 4th column) using the proposed adaptive soft-assignment (top row), re-ranked by visual word matching with geometric verification [11] (middle row), and re-ranked by the proposed visual word matching taking into account detected repttiles (bottom row). The green borders indicate correctly recognized images. The orange and cyan dots show the visual word matches and the inlier matches verified by the geometric verification, respectively, between the query image (1st column) and the best matching database image (2nd column). The matches are displayed on the 2nd and 3rd best matching images but their corresponding features are not shown in the query.

Fig. 14. **Examples of re-ranking by geometric verification on the San Francisco dataset.** See the caption of Figure 13 for details.

Examples of place recognition results demonstrating the benefits of the proposed re-ranking method are shown in Figures 13 and 14. The proposed method is more successful than the standard visual word matching because it generates more tentative matches on distinctive features while suppressing ambiguous matches on repetitive structures.

6.8 Evaluation on standard image retrieval datasets

We have also evaluated the proposed method for retrieval on the standard INRIA Holidays [8] and Oxford Buildings datasets [11], where performance is measured by the mean Average Precision (mAP). Here we consistently use $200,000$ visual vocabulary built from RootSIFT [54] features and $T = 5$ (different choices of T had small effect on the result). Results are summarized in Table 5 and demonstrate the benefits of the proposed approach over the standard BoVW baseline methods. However, the improvements by our method are less pronounced on this data. We believe this is because these datasets contain fewer repetitive structures than the place recognition imagery from urban areas (San Francisco, Pittsburgh) that is the main focus of this work.

TABLE 5
mAP on INRIA Holidays and Oxford Building datasets. Here we use 200K visual vocabulary built from RootSIFT [54] features and $T = 5$ (different choices of T had small effect on the result).

	tf-idf [11]	brst-idf [8]	SA tf-idf [16]	AA thr-idf
INRIA	0.7364	0.7199	0.7484	**0.7495**
Oxford	0.6128	0.6031	0.6336	**0.6565**

7 Conclusion

In this work we have demonstrated that repeated structures in images are not a nuisance but can form a distinguishing feature for many places. We treat repeated visual words as significant visual events, which can be detected and matched. This is achieved by robustly detecting repeated patterns of visual words in images, and adjusting their weights in the bag-of-visual-word representation. Multiple occurrences of repeated elements are used to provide a natural soft-assignment of features to visual words. The contribution of repetitive structures is controlled to prevent dominating the matching score. We have shown that the proposed representation achieves consistent improvements in place recognition performance in urban environments. In addition, the proposed method is simple and can be easily incorporated into existing large scale place recognition architectures. The open source implementation of the proposed method is available at [60].

Acknowledgments

We thank Relja Arandjelovic for discussions on descriptor compression. The authors were supported by JSPS KAKENHI Grant Number 24700161, EU FP7-SPACE-2012-312377 PRoViDE, TACR TA02011275 ATOM, the MSR-INRIA laboratory, EIT-ICT labs, the ANR project Semapolis, CityLabs@Inria and the ERC project LEAP.

References

[1] B. Aguera y Arcas, "Augmented reality using Bing maps." 2010, talk at TED 2010. [Online]. Available: http://www.videosift.com/video/TED-Augmented-reality-using-Bing-maps
[2] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM - FAB-MAP 2.0," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.
[3] D. Chen, G. Baatz *et al.*, "City-scale landmark identification on mobile devices," in *CVPR*, 2011.
[4] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *ECCV*, 2010.
[5] T. Quack, B. Leibe, and L. Van Gool, "World-scale mining of objects and events from community photo collections," in *Proc. CIVR*, 2008.
[6] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *CVPR*, 2007.
[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *ICCV*, 2007.
[8] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *CVPR*, 2009.
[9] H. Jegou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *CVPR*, 2007.
[10] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.
[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.
[12] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.
[13] A. Mikulik, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *ECCV*, 2010.
[14] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Descriptor learning for efficient retrieval," in *ECCV*, 2010.
[15] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *PAMI*, vol. 33, no. 1, pp. 117–128, 2011.
[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.
[17] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *PAMI*, vol. 32, no. 7, pp. 1271–1283, 2010.
[18] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *CVPR*, 2011.
[19] A. Irschara, C. Zach, J. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *CVPR*, 2009.
[20] Y. Li, N. Snavely, and D. Huttenlocher, "Location recognition using prioritized feature matching," in *ECCV*, 2010.
[21] J. Philbin, J. Sivic, and A. Zisserman, "Geometric latent dirichlet allocation on a matching graph for large-scale image datasets," *IJCV*, 2010.
[22] A. Torii, J. Sivic, and T. Pajdla, "Visual localization by linear combination of image descriptors," in *Proceedings of the 2nd IEEE Workshop on Mobile Vision, with ICCV*, 2011.
[23] P. Turcot and D. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problem," in *WS-LAVD, ICCV*, 2009.
[24] A. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *ECCV*, 2010.
[25] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *CVPR*, 2009.
[26] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large-scale image search," in *ECCV*, 2008.
[27] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *CVPR*, 2011.
[28] F. Schaffalitzky and A. Zisserman, "Automated location matching in movies," *CVIU*, vol. 92, pp. 236–264, 2003.
[29] C. Schmid and R. Mohr, "Local greyvalue invariants for image retrieval," *PAMI*, vol. 19, no. 5, pp. 530–534, 1997.
[30] J. Hays, M. Leordeanu, A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in *ECCV*, 2006.

[31] T. Leung and J. Malik, "Detecting, localizing and grouping repeated scene elements from an image," in *ECCV*, 1996.
[32] M. Park, K. Brocklehurst, R. Collins, and Y. Liu, "Deformed lattice detection in real-world images using mean-shift belief propagation," *PAMI*, vol. 31, no. 10, pp. 1804–1816, 2009.
[33] F. Schaffalitzky and A. Zisserman, "Geometric grouping of repeated elements within images," in *BMVC*, 1998.
[34] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert, "Detecting and matching repeated patterns for automatic geo-tagging in urban environments," in *CVPR*, 2008.
[35] J. Pritts, O. Chum, and J. Matas, "Detection, rectification and segmentation of coplanar repeated patterns," in *CVPR*, 2014.
[36] D. Hauagge and N. Snavely, "Image matching using local symmetry features," in *CVPR*, 2012.
[37] C. Wu, J. Frahm, and M. Pollefeys, "Detecting large repetitive structures with salient boundaries," in *ECCV*, 2010.
[38] P. Muller, G. Zeng, P. Wonka, and L. Van Gool, "Image-based procedural modeling of facades," *ACM TOG*, vol. 26, no. 3, p. 85, 2007.
[39] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of building facades using procedural shape priors," in *CVPR*, 2010.
[40] C. Wu, J.-M. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," in *CVPR*, 2011.
[41] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter," in *ICCV*, 2009.
[42] P. Doubek, J. Matas, M. Perdoch, and O. Chum, "Image matching and retrieval by repetitive patterns," in *ICPR*, 2010.
[43] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, 2013.
[44] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, 1988.
[45] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Lp-norm idf for large scale image search," in *CVPR*, 2013.
[46] S. Katz, "Distribution of content words and phrases in text and language modelling," *Natural Language Engineering*, vol. 2, no. 1, pp. 15–59, 1996.
[47] A. Pothen and C.-J. Fan, "Computing the block triangular form of a sparse matrix," *ACM Transactions on Mathematical Software*, vol. 16, no. 4, pp. 303–324, 1990.
[48] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[49] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable near identical image and shot detection," in *Proc. CIVR*, 2007.
[50] T. Pylvanainen, K. Roimela, R. Vedantham, J. Itaranta, and R. Grzeszczuk, "Automatic alignment and multi-view segmentation of street view data using 3d shape prior," in *3DPVT*, 2010.
[51] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited," in *BMVC*, 2012.
[52] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP*, 2009.
[53] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.
[54] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.
[55] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *PAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.
[56] H. Jegou, M. Douze, and C. Schmid, "Packing bag-of-features," in *ICCV*, 2009.
[57] O. Chum, J. Matas, and S. Obdrzalek, "Enhancing RANSAC by generalized model optimization," in *ACCV*, 2004.
[58] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
[59] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *CVPR*, 2005.
[60] http://www.ok.ctrl.titech.ac.jp/∼torii/project/repttile/.

Akihiko Torii Akihiko Torii received a Master degree and PhD from Chiba University in 2003 and 2006, respectively, for his research on omnidirectional vision. He then spent four years as a post-doctoral researcher in Centre for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. Since 2010, he is an assistant professor at the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests include large-scale 3D reconstruction and place recognition.

Josef Sivic Josef Sivic received a degree from the Czech Technical University, Prague, in 2002 and PhD from the University of Oxford in 2006. His thesis dealing with efficient visual search of images and videos was awarded the British Machine Vision Association 2007 Sullivan Thesis Prize and was short listed for the British Computer Society 2007 Distinguished Dissertation Award. His research interests include visual search and object recognition applied to large image and video collections. After spending six months as a postdoctoral researcher in the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, he currently holds a permanent position as an INRIA researcher at the Departement d'Informatique, Ecole Normale Superieure, Paris. He has published over 50 scientific publications and serves as an Associate Editor of the International Journal of Computer Vision. He has been awarded an ERC Starting grant in 2013.

Masatoshi Okutomi Masatoshi Okutomi received a B.Eng. degree from the Department of Mathematical Engineering and Information Physics, the University of Tokyo, Japan, in 1981 and an M.Eng. degree from the Department of Control Engineering, Tokyo Institute of Technology, Japan, in 1983. He joined Canon Research Center, Canon Inc., Tokyo, Japan, in 1983. From 1987 to 1990, he was a visiting research scientist in the School of Computer Science at Carnegie Mellon University, USA. In 1993, he received a D.Eng. degree for his research on stereo vision from Tokyo Institute of Technology. Since 1994, he has been with Tokyo Institute of Technology, where he is currently a professor in the Department of Mechanical and Control Engineering, the Graduate School of Science and Engineering.

Tomas Pajdla Tomas Pajdla received the MSc and PhD degrees from the Czech Technical University in Prague. He works in geometry and algebra of computer vision and robotics with emphasis on nonclassical cameras, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, noncentral camera models generated by linear mapping, generalized epipolar geometries, to developing solvers for minimal problems in structure from motion and to solving image matching problem. He coauthored works awarded prizes at OAGM 1998 and 2013, BMVC 2002 and ACCV 2014. He is a member of the IEEE. Google Scholar: http://scholar.google.com/citations?user=gnR4zf8AAAAJ